QoE-driven Cache Placement for Adaptive Video Streaming: Minding the Viewport

Othmane Belmoukadam, Chadi Barakat

QoE-driven Cache Placement for Adaptive Video Streaming: Minding the Viewport

Othmane Belmoukadam
Université Côte d'Azur, Inria, France
othmane.belmoukadam@inria.fr

Chadi Barakat
Université Côte d'Azur, Inria, France
chadi.barakat@inria.fr

***Abstract*—To handle the increasing demand for video streaming, ISP's and service providers use edge servers to cache video content to reduce the rush on their servers, balance the load between them and over the network, and smooth out the traffic variability. The dynamic adaptive streaming over HTTP protocol (DASH) makes videos available in multiple representations, and end-users can switch video resolution as a function of their network conditions and terminal display capacity (e.g., bandwidth, screen resolution). In this context, we study a viewport-aware caching optimization problem for dynamic adaptive video streaming that appropriately considers the client viewport size and access speed, the join time, and the characteristics of videos. We formulate and study the proposed optimization problem as an Integer Linear Program (ILP) that balances minimal join time and maximal visual experience, subject to the cache storage capacity. Our framework sheds light on optimal caching performance. Our proposed heuristic provides guidelines on the videos, and the representations of each video, to cache based on the video popularity, its encoding information, and the distribution of end-user display capacity and access speed in a way to maximize the overall end-user QoE.**

I. Introduction

The internet hosts plenty of services of all categories putting considerable pressure on its infrastructure, with internet video traffic being unavoidably the nightmare of operators. It is expected that by 2023, video traffic will account for 73% of the global mobile data traffic [1]. Lately, with the pandemic and the mobility restrictions, real-time entertainment based on streaming video and audio has become even more critical and has accelerated this continued growth [2]. End-users expect the best quality and can be frustrated by any service interruption, hence resulting in considerable economic losses for providers. In light of this rapid growth and increased economic impact, internet service providers feel more pressure to optimize their networks to meet the expectations of their end-users. For example, to prioritize or load balance traffic efficiently, ISP's need information on end-users QoE rather than just capturing the network Quality of Service (QoS). But, video QoE is dependent on the content itself (i.e., the video bitrate and resolution) and on the application-level QoS metrics such as the start-up delay, the duration of stalls, and the resolution switches [3]–[5]. It also depends on the viewport size [6], which can be defined as the number of pixels, both vertically and horizontally, on which the video is displayed. However, in the era of data encryption, all these metrics impacting the end-user QoE can be hard to infer.

Caching is another solution emerging through the surface. The main question is how to select the appropriate video to cache to maximize the overall users' QoE without exceeding the cache storage capacity. Several papers in the literature tackle the caching aspect for multimedia, in particular for video content (Sec., II). However, in the light of diverse and advanced equipment, the limitation of existing caching schemes and QoE-driven optimizations is overlooking the end-user display capacity. Studies have shown that different viewport resolutions have different bandwidth requirements even for the same QoE level [6]. On the same topic, we were able to leverage the existing relationship between screen resolution, video resolution, and QoE to formulate a resource allocation problem that maximizes the QoE for a set of users streaming videos over the same bottleneck link [7]. In this work, we propose a new cache placement optimization framework for adaptive video streaming that jointly accounts for the impact of end-user display capacity and video characteristics (e.g., encoding bitrate and popularity) in addition to the internet access speed. In plan, we formulate the optimal cache placement problem as an Integer Linear Program (ILP) aiming to maximize the average QoE over a set of users with a constraint on the cache storage capacity. The optimal solution, using CPLEX [8], can find the best selection of videos and representations to cache, ensuring minimal join time and maximal visual experience. Further, we develop a practical greedy caching heuristic using the optimal placement's footprint, offering a near-optimal performance in terms of average QoE per request. Overall, the main contributions of our paper are as follows; (i) We formulate the optimal cache placement problem for adaptive video streaming, the proposed framework leverages the users' viewport size heterogeneity and allocates the cache storage based on an objective function reflecting the QoE relation to the video content (bitrate), the application-level QoS (join time), the viewport size and the access speed distribution; (ii) We propose a near-optimal heuristic called $QoEscoreMax$ to solve for the optimization problem in a greedy way. We introduce a metric called $QoEscore$ to rank video representations and decide about caching them or not. This metric incorporates the expected QoE resulting from caching a particular representation of a certain video; (iii) We conduct simulations with multiple settings and show that our heuristic outperforms legacy caching strategies in terms of QoE gain.

II. RELATED WORK

Caching is an attractive solution for handling the increasing demand for video content. In particular, mobile edge caching (MEC) leverages storage capacity within the network to host popular multimedia content, easing video traffic delivery, smoothing its variability, and reducing congestion and access delay [9], [10]. Authors in [11] discuss thoroughly the benefits and limitations of caching content at the wireless edge and the needs to be considered for designing cache placement strategies. They also introduce methods to predict the popularity distribution and user preferences. Always in the wireless context, researchers have proposed the use of small cells called "helpers" to add caching functionality at the cellular access. The femtocache approach incorporates a wireless distributed caching network that assists the base station by handling requests of popular files that have been cached, thus minimizing the download delay of users [12], [13]. The work in [14] formulates a joint routing and caching problem while considering the bandwidth capacity constraints of the small cell base stations (SBS), aiming to maximize the fraction of content requests served locally by the deployed SBS's. Sengupta et al. propose an architecture to identify popular multimedia content by proactively pushing it close to the edge of the wireless network, thereby alleviating backhaul load [15].

To the best of our knowledge, the literature is still missing a study that accounts for the viewport resolution and its heterogeneity across the viewing users when addressing video caching. This constitutes the main focus of our paper.

III. FRAMEWORK AND SYSTEM MODEL

A. Framework

We consider a single edge cache scenario as depicted in Fig. 1. The origin server stores a catalog of video files, each of which is encoded into different representations. At the access, we have an edge server able to prefetch video files and cache them in advance. In general, edge servers are geographically closer to the users; the origin server leverages their caching storage to push popular content to the network edge during the off-peak hours, reducing the load on the origin server and resulting in more optimized delay and more convenient user experience. Usually, content providers put in place several edge servers to be as close as possible to different end-users, and one user can connect to several edge servers at a time. However, in this first study and to confirm the sound of our approach, we consider the case of one edge server. This assumption is similar to considering end-users able to connect to one edge server [16], which is also equivalent to optimize for each edge server individually.

In our context, whenever a client wants to play a video, it sends via its DASH client a request to the origin server, which gets redirected to the closest edge server, delivering back the highest video representation available and supported by the client network connection. If multiple representations of the requested video are available, the edge server will deliver back the one affordable by the client connection and her terminal display capacity. Usually, when no representation is found on the edge server, the user request is served directly by the origin server, delivering the best video representation affordable by the bottleneck link between the end-user and the origin server.

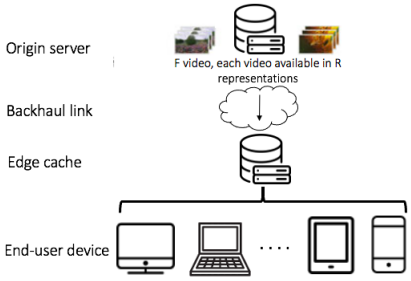


Fig. 1: Framework description

B. System model

We consider a catalog of $\mathcal{F}$ video files available at the origin server. Each video $f \in \mathcal{F}$ is available in $\mathcal{R}$ representations, such that $\forall\, r \in \mathcal{R}$ and $\forall\, f \in \mathcal{F}$, $B_{f,r}$ is the encoding bitrate of the representation r of video f. Moreover, we consider the $\mathcal{R}$ representations of each video to be ranked in increasing order of bitrates such that $B_{f,r-1} \leq B_{f,r}, 1 < r \leq |\mathcal{R}|$. We will also assume that all videos have the same duration T. Such assumption has often been adopted in the literature for simplicity and with no loss of generality [16]. Let E_c be the cache of the edge server, and let S_c be its available cache storage capacity in Bytes. On the other hand, let $\mathcal{D}$ denote the set of users' devices that request videos and that are eligible to communicate with E_c. Each $d \in \mathcal{D}$ reaches E_c with a download rate capacity equal to c_d which we assume to be fully dedicated to the video streaming of the device. This capacity, already captured by the DASH client, can be approximated from past delivered chunks. Moreover, we denote by v_d the viewport resolution of device d. As for content popularity distribution, we assume it to be stationary over the optimization period, and we consider requests to be independent of each other following the well-known Independent Reference Model. We denote by P_f the popularity of video f and we normalize it in such a way that it becomes equal to the probability that any request issued by any device $d \in \mathcal{D}$ hits video f independently of the other requests [16].

We aim for a cache placement decision to be made by the origin server, or any other controller, in a discrete-time manner. In terms of end-user viewport size (v_d), content providers have access to this information as it is communicated between the DASH client and the DASH server. Such data can be inferred using machine learning with features calculated on the video encrypted traffic [17].

C. QoE modeling

The video QoE models in state of the art focus mainly on application-level QoS metrics. However, the viewport size is also an important factor impacting the visual experience.

1) From bitrate to QoE: We capture the relationship between the viewport size and the selected video resolution (e.g., encoding bitrate) and the latter's impact on the QoE. We leverage an exponential QoE model calibrated offline using an open-source dataset [6]. This model maps the encoding bitrate, z_{BR}, with the perceived user experience, z_{MOS}, for a set of standard viewport sizes. Using the dataset in [6], we extrapolate a vector Z where each entry has two values (z_{BR},z_{MOS}) then use the mean square error method to fit curves according to Equation (1). In plain, the β_{v_d} derived with curve fitting describes the shape of the function for the viewport resolution of the device d, and QoE_{max} is the maximum anticipated QoE value.

$$QoE_{v_d} \underset{exp}{=} QoE_{max}(1 - e^{-\beta_{v_d} x}). \tag{1}$$

2) From join time to QoE: We also account for the join time (initial delay), which is the time it takes the video to start playing out. According to authors in [18], users start abandoning the video session after 2 seconds of join time, and 80% of them leave the session when their join time exceeds 60 seconds. We consider a logarithmic model for the impact of the join time on the QoE as proposed in [19]. Equation (2) provides a version of this model fitted by the authors of [19] using a crowd-sourced dataset of YouTube video streaming. In this equation, $join_d$ is the join time experienced by device d, which can be set to the time needed to fill up the playout buffer on the device. Equation (3) provides an estimation of this time using the encoding bitrate of the representation r of video f, $B_{f,r}$, the playout buffer size in seconds, δ, and the user connection speed c_d.

$$QoE_{join_d} = -0.963 \times \log(join_d + 5.381) + 5, \tag{2}$$

$$join_d = \frac{\delta \times B_{f,r}}{c_d}. \tag{3}$$

IV. Viewport aware optimal cache placement

The viewport-aware cache placement problem for adaptive video streaming can be described as follows. Given a catalog of videos and the different available representations, the video popularity distribution, the end-user maximum download speed, the end-user viewport resolution, select the most rewarding set of video representations to be cached such that the total system utility is maximized as constrained to cache storage capacity.

A. Utility function

For simplicity and without loss of generality, we consider a caching system where a representation of a video file is either fully cached or not cached at all. We assume that any representation can be played out on any viewport and that devices have different viewport resolutions and internet connection speeds. Further, any representation exceeding the resolution of the viewport brings the maximum level of QoE [20]. In this context representing better the reality, the decision on the best representations to cache becomes more complex to solve. To reach an optimal solution, we first start by introducing a binary variable $\alpha_{f,r}$ for the action of caching a representation or not. We then complement it with another binary variable per device d called $\gamma^d_{f,r}$ that specifies which representation of video f is served by the cache to device d in case one or more representations of the video are available in the cache. Otherwise, the request is served by the origin server.

$$\alpha_{f,r} = \begin{cases} 1, & \text{if } file(f,r) \text{ } cached \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

$$\gamma^d_{f,r} = \begin{cases} 1, & \text{if } file(f,r) \text{ } served \text{ } to \text{ } d \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

We define the QoE-driven utility function for a request issued by device d as the average QoE reward overall videos of the catalog while conditioning on the viewport resolution and the device's connection speed d. We write it as a weighted sum of the two QoE functions defined in Section III-C:

$$Qgain_d = \sum_{f \in \mathcal{F}} P_f \sum_{r \in \mathcal{R}} \gamma^d_{f,r} \times (a \times QoE_{v_d} + b \times QoE_{join_d}). \tag{6}$$

(a, b) are system parameters that can be tuned to adjust the importance of each QoE aspect.

B. Problem formulation

The QoE-driven cache placement problem for adaptive streaming can be formulated as an Integer Linear Program (ILP) in the following way:

$$\max_{\alpha,\gamma} \sum_{d \in \mathcal{D}} Qgain_d \tag{7}$$

$$\text{subject to: } \sum_{f \in \mathcal{F}} \sum_{r \in \mathcal{R}} \alpha_{f,r} \times B_{f,r} \times T \leq S_c, \tag{8}$$

$$\sum_{r \in \mathcal{R}} \gamma^d_{f,r} \leq 1, \ \forall f \in \mathcal{F}, \ \forall d \in \mathcal{D}, \tag{9}$$

$$\gamma^d_{f,r} \leq \alpha_{f,r}, \ \forall f \in \mathcal{F}, \ \forall r \in \mathcal{R}, \ \forall d \in \mathcal{D}, \tag{10}$$

$$\alpha_{f,r} \in \{0, 1\}, \tag{11}$$

$$\gamma^d_{f,r} \in \{0, 1\}. \tag{12}$$

In this problem formulation, the objective is to maximize the overall QoE reward summed over the set of devices as defined in (7) and (6), while considering the network conditions, the video characteristics (e.g., popularity and encoding bitrate) and the end-user viewport size. The constraint in (8) represents the cache size constraint, with $B_{f,r} * T$ being the part of the cache occupied if we cache file (f, r). The constraint in (9) makes sure that each device can only download one representation per cached video. The constraint in (10) establishes the relationship between the two decision variables such that a video representation can be served only if it is already cached. Finally, the constraints in (11) and (12) define the binary decisions of caching and serving, respectively.

C. QoEscoreMax

We present a greedy heuristic named $QoEscoreMax$ based on the notion of $QoEscore$. The $QoEscore$ is a new metric we introduce to calculate for each video representation the QoE gain that would result from caching it, summed over the set of devices. Following the same reasoning as in Eq. (6), we write:

$$QoEscore_{f,r} = \sum_{d \in \mathcal{D}} P_f * (a * QoE_{v_d} + b * QoE_{join_d}).$$

To further account for the cache space occupied by the video file, we normalize the score by the square of its volume in Bytes, $B_{f,r} * T$ [1]. We use the normalized score to rank representations in decreasing order of QoE gain. The $QoEscoreMax$ algorithm caches files having the highest $QoEscore$ in the limit of the cache storage.

By studying the footprint of the optimal solution as solved by CPLEX, we found out that depending on the network conditions and the viewport resolution distribution. We might only need one representation to hit the optimal. Following this optimal footprint, we update $QoEscoreMax$ to limit the number of representations per video. Overall, we iterate over the $QoEscore$ ranked list with three possible options: either replacing, adding, or simply skipping. For instance, for file (f, r), if we do not have the previous representation cached (e.g., $cached_{f,r-1} = 0$), we add the (f, r) representation directly to the cache while increasing the cache occupancy, otherwise, a $delatQoEgain$ is computed between the two cache states: (1) the new representation replaces the previous one, and (2) the new representation is skipped. A value of $delatQoEgain$ positive means replacing the previous representation is beneficial and so is taken. Otherwise, no action is taken until the following representation of video f is found in the ranked list of $QoEscore$. This heuristic is detailed in Algorithm 1.

Result: $Cached\ -\ binary\ placement\ list\ (\mathcal{F}, \mathcal{R})$
$QoEscore(\mathcal{F}, \mathcal{R}), S_c, cache_{occ}, T, B\ (\mathcal{F}, \mathcal{R})$
while $B_{(f,r)} * T + cache_{occ} \leq S_c$ **do**
  if $cached_{(f,r-1)} = 0$ **then**
    $cached_{(f,r)} = 1$
    $cache_{occ} = cache_{occ} + B_{(f,r)} * T$
  else
    if *deltaQoEgain(file(f,r), file(f,r-1))* ≥ 0 **then**
      $cached_{(f,r)} = 1$
      $cached_{(f,r-1)} = 0$
      $cache_{occ} = cache_{occ} - B_{(f,r-1)} * T$
      $cache_{occ} = cache_{occ} + B_{(f,r)} * T$
    end
  end
  $(f, r) =$*QoEscore*$.next_{key}$
end

Algorithm 1: QoEscoreMax

[1]The normalization by the square of the volume was shown empirically to provide better results than the normalization by the volume itself.

In terms of complexity, assuming the lookup operations of O(1) (i.e., hash maps), the running time of the proposed $QoEscoreMax$ greedy algorithm is O($(FR)^2D$), resulting in polynomial time complexity. The CPLEX provides the optimal solution using the branch-and-cut search. This method follows a search tree consisting of nodes representing a relaxed LP subproblem. Each subproblem can be solved using the SIMPLEX method with exponential time complexity.

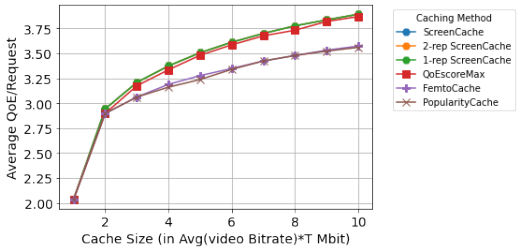
V. Performance evaluation

To assess the efficiency of our approach, we compare it to state of the art approaches such as popularity-based caching, which takes into consideration the video popularity [16], [21] and Femtocaching which minimizes the average download delay of video content [13]. Beside our heuristic $QoEscoreMax$, we study different variants of our optimal solution, in particular we show results for *(i)* $ScreenCache$ which does not put any limit on the number of cached representations per video, and *(ii)* $1 - rep - ScreenCache$ and $2 - rep - ScreenCache$ which limit the number of cached representations per video to maximum 1 ($\sum_{r \in \mathcal{R}} \alpha_{f,r} \leq 1$) and 2 ($\sum_{r \in \mathcal{R}} \alpha_{f,r} \leq 2$) representations, respectively. For popularity-based caching, we implement a greedy version called $PopularityCache$ that caches representations in increasing bitrate order using the popularity ranking. For $FemtoCache$, we use CPLEX to get its optimal solution leveraging the video popularity and the network conditions.

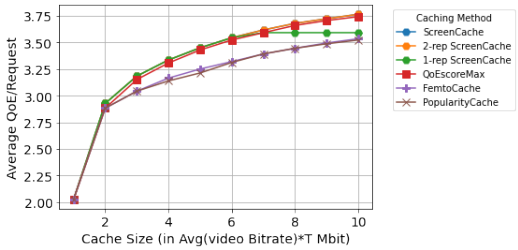
A. Simulation settings

We develop a numerical simulator in Python where videos are cached and QoE calculated according to Eq. (1), (2) and (3). We consider a network of 20 devices and sample the devices' viewports over a set of standard viewport sizes (420x240, 640x360, 850x480, 1280x720, 1920x1080). In terms of network access, we consider two scenarios depicting a case where users have either high download rates (from 10 to 18 Mbps) or poor/medium download rates (from 1 to 7 Mbps). As for video content, we consider a catalog of 20 videos of same duration $T = 60s$, each video is encoded in 7 representations with encoding bitrates (0.25, 0.55, 0.95, 1.5, 2.6, 5, 8 Mbps). We further assume that the popularity of the videos follows a Zipf distribution with parameter 0.56 [22]. Last, the storage capacity is varied as multiple of the average size of a video representation.

At this stage, we consider $a = b = 0.5$ in Eq. (6), such that the encoding bitrate and the join time have the same importance on the user experience. To compare the different caching strategies, we use the metric $AverageQoE/request$, representing the average perceived QoE over the set of devices and videos. Each request targeting a random video in the catalog will be potentially served by the cache given the selection of cached representations and following the process described in Sec. III. This metric, between 0 and 4.5 (maximum QoE), also includes the notion of hit/miss, as the cache misses will result in zero contribution to the QoE_{gain}. We do not consider



(a) Fast internet accesses



(b) Poor/Medium internet accesses

Fig. 2: Average QoE per request, uniform viewport distribution

the QoE of downloading from the origin server in case of a miss because we aim at optimizing the cache behavior independently of the internet backbone.

B. Simulation results

For space constraints, we only show results related to uniform viewport resolution distribution. Other viewport resolution distributions have demonstrated similar behavior as the one we will show next.

We start with the scenario of high access rates. We plot in Fig. 2(a) the $AverageQoE/request$ vs. cache capacity, viewports of the 20 devices were sampled uniformly. Here, the optimal $ScreenCache$ derived by the CPLEX results in the same QoE as $1-rep-ScreenCache$ and $2-rep-ScreenCache$, suggesting that one representation per video can strike the optimal. On the other hand, $FemtoCache$ and $PopularityCache$ perform similarly, and below the optimal, the reason is that $PopularityCache$ by proceeding in increasing order of bitrates ends up giving priority to the smallest representations, which results in almost the same behavior as the $FemtoCache$ scheme which tries to minimize the average file download delay. Meanwhile, $QoEscoreMax$ outperforms the previous two caching strategies and highlights a near-optimal performance. Thanks to using the $QoEscore$ metric, $QoEscoreMax$ favorites the most rewarding representations making possible the caching of other than the lowest representation if needed by some viewports and some access links. Large viewport resolutions with good internet access make schemes focusing on minimizing the file download delay less efficient than the optimal that cache directly those representations providing the maximum QoE gain.

In a second scenario, we consider devices with poor to medium internet access (i.e., part of devices cannot accommodate all representations). This scenario is more challenging as it requires caching a mix of representations depending on the access speed and the viewport. Here, one can expect $1-rep-ScreenCache$ to diverge from unlimited $ScreenCache$ as the cache size increases. As we can observe in Fig. 2(b), the $1-rep-ScreenCache$ scheme starts indeed diverging from the optimal as the cache size increases. Finding one representation per video that approximates well the optimal for all viewports is no longer possible as some accesses are slow and cannot accommodate high-quality representations. However, we can see that the $2-rep-ScreenCache$ keeps up and shows almost the same behavior as the unlimited optimal. To further understand this behavior, we analyze the footprint of $ScreenCache$; the optimal considers two representations for popular videos while holding the least popular videos to one representation. $QoEscoreMax$ sustains its good performance through the different viewport distributions.

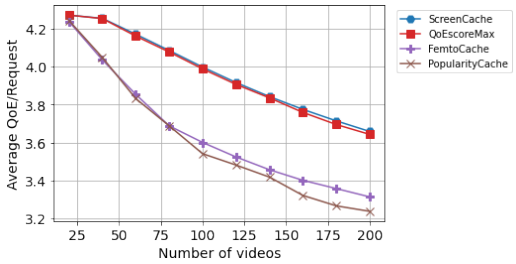


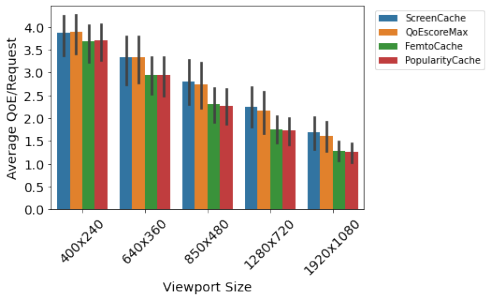
Fig. 3: QoE vs catalog size

Moreover, we test the behavior of our solution for a larger video catalog. We plot in Fig. 3 the average QoE per request for the scenario of good network conditions while scaling up the catalog size at a fixed cache size of ten times the average representation size ($10 * Avg(B_{f,r}) * T$ Mbits). Overall, the QoE value is negatively correlated with the catalog size for all caching strategies, making sense since the storage capacity remains the same and the pressure on the cache increases. However, the decline of $ScreenCache$ and $QoEscoreMax$ is slower than $FemtoCache$ and $PopularityCache$ as the former can better utilize the available storage by caching the most rewarding content directly rather than caching low-quality videos for unpopular content.

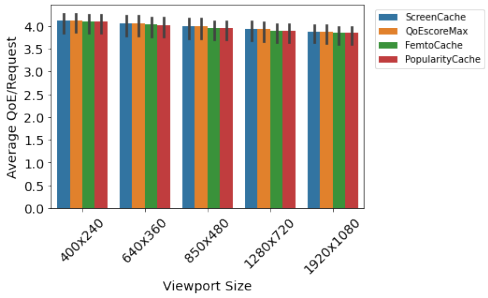
VI. Sensitivity Analysis

Here, we evaluate the impact of the QoE model parameters on the cache placement strategy. We study the impact of the balance between join time and bitrate, and show results for good network conditions and uniform viewport distribution.

1) Video bitrate over join time: In this part, we give more importance to QoE_{v_d} linking the bitrate to the QoE, with $a = 0.9$ and $b = 0.1$. We plot in Fig. 4(a) the average QoE per request (plus its standard deviation) for different viewport sizes and different caching schemes. Overall, we observe that the QoE decreases as we move toward larger viewports. Between the caching schemes, $ScreenCache$ and $QoEscoreMax$ result in almost the same QoE level per view-



(a) Bitrate over join time



(b) Join time over bitrate

Fig. 4: Average QoE per viewport

port resolution, while $FemtoCache$ and $PopularityCache$ fall behind, especially for large screens.

2) Join time over video bitrate: Here, we assume that the QoE model in Eq. (6) is mostly based on the join time by considering $a = 0.1$ and $b = 0.9$. Since QoE_{join_d} is negatively correlated with $join_d$ (Eq. (2) and (3)), one would expect the optimal solution to be selecting representations with smallest encoding rate as they reduce the join time. Differently speaking, the model now largely prefers the smoothness of the playout on the quality of the rendered resolution, which is closer in mind to existing placement schemes that seek to minimize the file download delay by caching first the low representations. We illustrate the obtained results in Fig. 4(b). When favoring the join time, and regardless of the viewport size, the different caching schemes converge to almost the same QoE level, thus, caching almost the same content.

VII. Conclusion and Future Work

In this work, we study a QoE-driven cache placement optimization for adaptive video streaming while accounting for the end-user viewport. We formulate the problem as an ILP and derive the optimal selection of videos and representations to be cached for different internet accesses and viewport size distributions. We also present $QoescoreMax$, a practical caching heuristic with near-optimal performance. Simulation results show that our solution strikes the trade-off between optimal QoE and efficient storage management. Moreover, they provide insights on video representations selection based on network conditions and the importance of the balance between join time and video bitrate. In good network conditions, one representation can lead to optimal QoE. For mild network conditions, the selection process has to account for the videos' popularity before adding another video representation.

We plan to extend this study by testing our solution in a cooperative caching scenario where coordination between multiple edge servers is enabled and orchestrated by a main controller, mainly in the context of Software Defined Networks.

References

[1] Ericsson, "Ericsson Mobility Report, June 2018," https://www.ericsson.com/assets/local/mobility-report/documents/2018/ericsson-mobility-report-june-2018.pdf, 2018.

[2] A. Feldmann, O. Gasser, F. Lichtblau, E. Pujol, I. Poese, C. Dietzel, D. Wagner, M. Wichtlhuber, J. Tapiador, N. Vallina-Rodriguez, O. Hohlfeld, and G. Smaragdakis, "The lockdown effect: Implications of the covid-19 pandemic on internet traffic," in *Proceedings of the ACM Internet Measurement Conference*, 2020.

[3] R. R. Pastrana-Vidal, J. C. Gicquel, C. Colomes, and H. Cherifi, "Sporadic frame dropping impact on quality perception," in *Proc. SPIE 5292, Human Vision and Electronic Imaging IX*, 2004.

[4] Y. Qi and M. Dai, "The effect of frame freezing and frame skipping on video quality," in *IEEE Multimedia Signal Process.*, 2006.

[5] A. K. Moorthy, L. K. Choi, A. C. Bovik, and G. D. Veciana, "Video quality assessment on mobile devices: Subjective, behavioral and objective studies,," in *IEEE J. Sel. Topics Signal Process.*, 2012.

[6] G. Cermak, M. Pinson, and S. Wolf, "The relationship among video quality, screen resolution, and bit rate," in *IEEE Transactions on Broadcasting*, 2011.

[7] O. Belmoukadam, M. J. Khokhar, and C. Barakat, "On accounting for screen resolution in adaptive video streaming: A qoe-driven bandwidth sharing framework," in *CNSM*, 2019.

[8] IBM, "ILOG CPLEX optimization studio." https://www.ibm.com/products/ilog-cplex-optimization-studio, 2020.

[9] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet of Things Journal*, 2018.

[10] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Transactions on Information Theory*, 2014.

[11] D. Liu, B. Chen, C. Yang, and A. F. Molisch, "Caching at the wireless edge: design aspects, challenges, and future directions," *IEEE Communications Magazine*, 2016.

[12] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Transactions on Information Theory*, 2013.

[13] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," in *IEEE INFOCOM*, 2012.

[14] K. Poularakis, G. Iosifidis, and L. Tassiulas, "Approximation algorithms for mobile data caching in small cell networks," *IEEE Transactions on Communications*, 2014.

[15] A. Sengupta, R. Tandon, and O. Simeone, "Cache aided wireless networks: Tradeoffs between storage and latency," in *Conference on Information Science and Systems*, 2016.

[16] Y. Jin, Y. Wen, and C. Westphal, "Optimal transcoding and caching for adaptive streaming in media cloud: an analytical approach," *IEEE Transactions on Circuits and Systems for Video Technology*, 2015.

[17] O. Belmoukadam and C. Barakat, "From encrypted video traces to viewport classification," in *CNSM*, 2020.

[18] S. S. Krishnan and R. K. Sitaraman, "Video stream quality impacts viewer behavior: Inferring causality using quasi-experimental designs," *IEEE/ACM Transactions on Networking*, 2013.

[19] T. Hossfeld, S. Egger, R. Schatz, M. Fiedler, K. Masuch, and C. Lorentzen, "Initial delay vs. interruptions: Between the devil and the deep blue sea," in *4th International Workshop on Quality of Multimedia Experience*, 2012.

[20] A. Mansy, M. Fayed, and M. Ammar, "Network-layer fairness for adaptive video streams," in *IFIP Networking Conference*, 2015.

[21] W. Li, S. M. A. Oteafy, and H. S. Hassanein, "Streamcache: Popularity-based caching for adaptive streaming over information-centric networks," in *International Conference on Communications*, 2016.

[22] C. Li, L. Toni, J. Zou, H. Xiong, and P. Frossard, "Qoe-driven mobile edge caching placement for adaptive video streaming," *IEEE Transactions on Multimedia*, 2018.